Optimal Element Placement in Conformal Beamforming

Richard Leahy[†] *Brian Jeffs*[†‡]

[†]Signal and Image Processing Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

[‡]Hughes Aircraft Company
7000 Village Drive
Buena Park, CA 90621

ABSTRACT

An important problem in the design of some array beamformers is to find the number of array elements and their spatial locations necessary to meet a given response. In this paper we present a new technique for simultaneous determination of the minimum number of array elements, their locations and the shading weights necessary to meet some response specified in terms of a set of linear inequality constraints. Results are presented for the design of optimally sparse linear and circular arrays for narrow band applications.

Introduction

In this paper we consider the problem of array element shading and placement for symmetric linear, planar and arbitrarily shaped 3D-arrays in narrow-band phased beamformer operation. To solve this problem we constrain the response of the beamformer using a set of linear inequalities. Using a large number of candidate array element locations and minimizing a cost function (which decreases as the number of nonzero weights, and hence array elements, decreases) over the constraint set yields the optimally sparse array.

Although a number of line array thinning approaches have been proposed [1,2], they are not applicable to arbitrary 3-D arrays, and do not guarantee optimally sparse arrays even in the 1-D case. The literature contains analysis of a number of unusually shaped arrays, including circular, spherical, cylindrical, and conformal configurations which follow the the shape of a supporting vehicle [3,4,5,6]. With these configurations it is often very difficult to determine efficient element placement; attempting to approximate equal spacing can cluster elements in areas which contribute little to array response, and thinning can become a trial and error proposition. Maximizing the target signal to noise ratio with respect to a known noise field [7], linear programming methods, or using a pattern search algorithm [6,8] can yield useful shadings for these arbitrary arrays, but can give no information on how many elements we need, or where they should be placed.

In the following we describe a design algorithm which can be applied to any symmetric array problem and will find the minimum number of necessary elements. The problem is formulated in section 2; in section 3 we state two theorems on which our algorithm is based and briefly describe the algorithm. The method is then demonstrated in application to the design of linear and planar arrays, the method extends directly to the design of symmetric 3D arrays.

Optimal Design of Narrowband Arrays

System Constraints

To set up a system for beamforming design we take M samples of the upper and lower response bounds from a dense enough grid on an enclosing sphere to control sidelobe leakage. Let $\mathbf{s}_i$ be the vector of direction cosines to the point on the sphere where the upper and lower response constraints, b_{u_i} and b_{l_i}, are sampled and let $\mathbf{s}_0$ be the vector direction cosines of the desired maximum response angle (MRA). A large number of potential array element locations, $\mathbf{r}_j$, $j=1,..N$, are defined with unknown shading weights a_j. We require symmetry about the origin to ensure a real response value:

$$\mathbf{r}_j = -\mathbf{r}_{N-j-1}, \text{ and } a_j = a_{N-j-1} \tag{1}$$

therefore we need only solve for $N/2+1$ coefficients in $\underline{a}$. The response, $\underline{d}$, at the constraint points is then given by the cosine transform, $\underline{d}=\mathbf{A}\underline{a}$, where the matrix $\mathbf{A}$ has elements:

$$\mathbf{A}_{ij} = \frac{2}{N}\cos\left[\mathbf{r}_j \cdot (\mathbf{s}_i-\mathbf{s}_0)\omega/c\right], \quad \text{for } \begin{matrix} i=1,\cdots M \\ j=1\cdots N/2+1 \end{matrix} \tag{2}$$

where ω is the band center radian frequency and c is the wave propagation speed. The final complex element weight for the beamformer is then

$$\gamma_j = a_j \mathbf{e}^{-\mathrm{j}\frac{\omega}{c}(\mathbf{r}_i \cdot \mathbf{s}_0)} \tag{3}$$

As we shall see below, the algorithm is based on a simplex search performed over a non-negative set. Let a_j be the computed shade for the jth element, with $a_j = a_j^+ - a_j^-$ where $a_j^+, a_j^- \geq 0$; we may then obtain positive or negative shade values while using the non-negative vectors $\underline{a}^+$ and $\underline{a}^-$ in the algorithm. We use a similar formulation to that applied in linear programming [9] and replace the set of inequality constraints with equality constraints by intro-

22ACSSC-12/88/0803 $1.00 © 1988 MAPLE PRESS

ducing non-negative slack and surplus vectors z^+ and z^- of length M. The constraints may then be rewritten in the form:

$$\mathbf{H}x = b \; where \; \mathbf{H} = \begin{bmatrix} \mathbf{A} & -\mathbf{A} & \mathbf{I} & 0 \\ \mathbf{A} & -\mathbf{A} & 0 & -\mathbf{I} \\ 1 & 1 & 0 & 0 \end{bmatrix}, \; x = \begin{bmatrix} a^+ \\ a^- \\ z^+ \\ z^- \end{bmatrix}, \; b = \begin{bmatrix} b_u \\ b_l \\ c \end{bmatrix} \quad (4)$$
$$x \geq 0$$

Any vector which is a feasible solution to (4) yields a feasible solution to the constraint $b_l \leq \mathbf{A}a \leq b^u$, with $a = (a^+ - a^-)$. The last row of $\mathbf{H}$ is added to constrain the sum of shade absolute values where c can be adjusted to improve beamformer stability and array gain relative to a noise field.

Optimally Sparse solutions

To find the optimally sparse solution we must optimize a suitable cost function over the set of linear constraints. The problem can be expressed as the nonlinear mathematical program:

$$\min_{a} \; f(a) = \sum_{i=1}^{N} \mathrm{I}(a_i) \quad such \; that \quad b_l \leq \mathbf{A}\,a \leq b_u \quad (5)$$
$$\sum_{i=1}^{N} |a_i| \leq c$$

$$where \; \mathrm{I}(a_i) = \begin{cases} 1 & a_i \neq 0 \\ 0 & a_i = 0 \end{cases}$$

The cost function $f(a)$ penalizes any array element requiring a nonzero shading. Clearly an element with zero shade is not necessary and hence minimization of (5) will yield an optimally sparse array. This procedure may be viewed as either array thinning or element placement, depending on how densely the original elements are arranged.

As an optimization problem, (5) is particularly difficult to solve. We are plagued with numerous local minima, and $f(a)$ is discontinuous and has zero gradient except at the discontinuities. In an effort to overcome these limitations, we propose an approach to the minimum order problem based on generalized, linearly constrained l_p optimization.

The algorithms for minimum order optimization discussed below are based on the related nonlinear program

$$\min_{a} \; g_q(a) = \sum_{i=1}^{N} |a_i|^{\frac{1}{q}} \quad s.t. \quad b_l \leq \mathbf{A}a \leq b_u, \; q > 1 \quad (6a)$$
$$\sum_{i=1}^{N} |a_i| \leq c$$

The problem may be rewritten in canonical form using (4) as

$$\min_{x \geq 0} \; g_q'(x) = \sum_{i=1}^{N} |a_i^+|^{\frac{1}{q}} + |a_i^-|^{\frac{1}{q}} \quad s.t. \quad \mathbf{H}x = b, \; q > 1 \quad (6b)$$
$$a_i^{+T} a^- = 0$$

Note that the cost function in (6a) is the sum of the q^{th} roots of each element of the vector a. Since the q^{th} root of any positive number converges towards unity as q increases and any root of zero is zero, it follows that in the limit as $q \rightarrow \infty$, $g_q(a) \rightarrow f(a)$. The motivation for using $g_q(a)$ in place of $f(a)$ is to develop a numerically stable algorithm. A theorem is given below stating conditions under which an optimal solution to (6a) is also an optimal solution to (5).

Theorems

To motivate the algorithm development we give two theorems, proofs may be found in [10,11]. We borrow the following definition from the related problem of linear programming: a basic feasible solution (BFS) to a set of K equality constraints is one which satisfies those constraints and has at most K non-zero components.

Theorem 1A

If a solution to (6b) exists, then a basic feasible solution exists.

Theorem 1B

If a globally optimal solution to (6b) exists, then a globally optimal basic feasible solution exists. Furthermore, this solution is a globally optimal solution to (6a).

The important consequence of this theorem, is that problems (6a) and (6b) are equivalent and hence the optimal solution may be found by searching the set of BFSs to (6b). Note that this result is very similar to the fundamental theorem of linear programming, and results from the concave nature of $g_q(a)$ for $a \geq 0$. Since the optimal solution must be a BFS of the constraints in (6b), we can restrict our search to this finite set of points, i.e. the solution may be found using a simplex algorithm.

The second theorem gives a theoretical bound on the value of q for which an optimal solution to (6a) is an optimal solution to (5).

Theorem 2

Let S denote the set of all BFSs to (6b). If all BFSs are bounded, then let Ω denote the maximum and ϵ the minimum (non-zero) values of all elements from the vectors in S. If V is the set of all globally optimal solutions to (5) with $r = f(x)$ for any $x \in V$, and U is the set of all globally optimal solutions to (6a), then $U \subset V$ if $q \geq q_1$ where

$$q_1 = \frac{\log\left(\frac{\Omega}{\epsilon}\right)}{\log\left(\frac{r+1}{r}\right)} \quad (7)$$

This result gives us a conservative upper bound on q for any problem for which the BFSs are bounded. This bound is difficult to compute and in practice a value of q is chosen using estimates of Ω and ϵ based on the physical limits of the system.

Search Algorithms

A full description of the optimization algorithms is given in [10]. In the following we will briefly outline the procedures.

An initial BFS may be found using an initializing algorithm identical to the first stage of the conventional linear programming algorithm [9]. Associated with this is a 'simplex tableau' which may be found by performing successive pivoting operations on the original constraint tableau. We label the indices of the BFS vector as either 'in' or 'out' of the basis, depending on whether the corresponding element is zero or not. The situation becomes somewhat more complex for degenerate systems [12], discussion of this case is omitted here for reasons of space.

One can move from one BFS to another by pivoting operations which move one variable into, and another variable out of, the basis. Two BFSs are said to be adjacent if one can move from one to the other using a single pivot. Using this definition we can construct a connected graph from the BFSs. The goal of our algorithm is then to find the global minimum of the cost function on this graph. Thus at each stage of the algorithm one performs a single pivot, moving from one BFS to another so that the cost at each iteration is monotonically decreasing by selecting an adjacent BFS of lower cost. Convergence is then defined as the point in the sequence of BFSs for which no adjacent BFS is of equal or lower cost.

This algorithm has been demonstrated to yield sparse solutions as shown below. The effective mechanisms by which this achieved are (i) to have as many of the slack and surplus variables in the basis as possible, since then the minimum number of components of $\underline{a}$ will be nonzero; and (ii) to find degenerate solutions in which elements of $\underline{a}$ are in the basis but have zero values.

Although the algorithm usually produces acceptably sparse solutions, it cannot ensure a globally optimal result and may terminate at a local optimum on the graph of BFSs. A global optimization can only be found if the algorithm allows one to escape from local minima. To see how this may be achieved, consider the sequence of BFSs obtained using the algorithm above. Each step is dependent only on the most recent element in the sequence. Thus if one chooses the next element of the sequence randomly from the adjacent BFSs of the current element, the resulting sequence must be a Markov chain defined on the graph of BFSs. The key to obtaining global convergence is to define an appropriate updating rule such that one can obtain a homogeneous Markov chain [13]. Using this concept, we have developed a simulated annealing algorithm which has guaranteed asymptotic convergence to a global minimum provided an appropriate annealing schedule [13] is employed. As shown below, in practice one can obtain solutions of lower order than the local search algorithm in finite time using simulated annealing.

Computed results

Linear Array Design

As an example of the design of sparse arrays we compare our algorithm with a linear equispaced array designed using Chebyschev approximation. The maximally sparse arrays were designed starting with 120 array elements equispaced with .1m separation for a sonar system operating at a wavelength of 1.5m. The response of this array, using unity magnitude shading (see equ (3)) is shown in Fig. 1. The simplex algorithm was then applied to the system with the constraint that the mainlobe width and

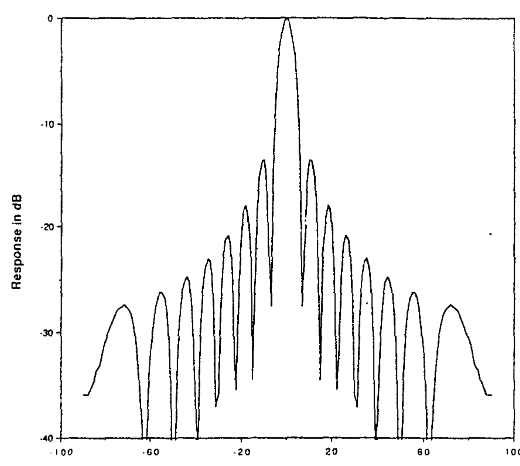

Figure 1: Response of 120 element linear equispaced array with unit magnitude shading.

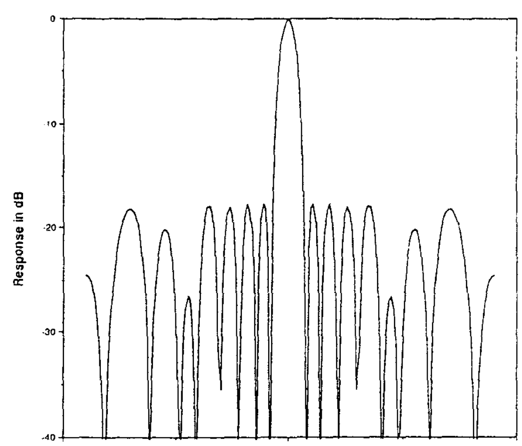

Figure 2: Response of linear array after thinning from 120 to 11 elements using local simplex search.

maximum sidelobes were identical to those of Fig. 1. The simplex search reduced the number of elements necessary for these constraints to 11. Using the same constraints with the stochastic search reduced the number of elements to 9. The resulting responses are shown in Figs. 2 and 3. For comparison, an array with half wavelength spacing was designed using Chebyschev approximation with identical constraints to those for Figs. 2 and 3. In this case a total of 15 elements were necessary to meet the constraint, Fig. 4. The resulting element locations for Figs. 2-4 are shown in Fig. 5.

These results clearly indicate the ability of the algorithm to design sparse linear arrays, where the use of unequal element spacing results in a reduction in the required number of array elements.

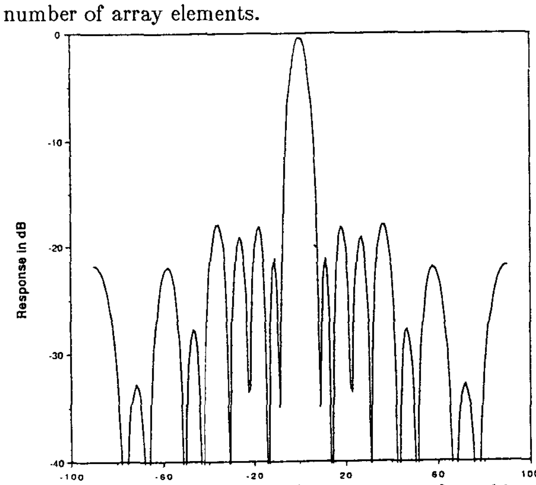

Figure 3: Response of linear array after thinning from 120 to 9 elements using stochastic simplex search.

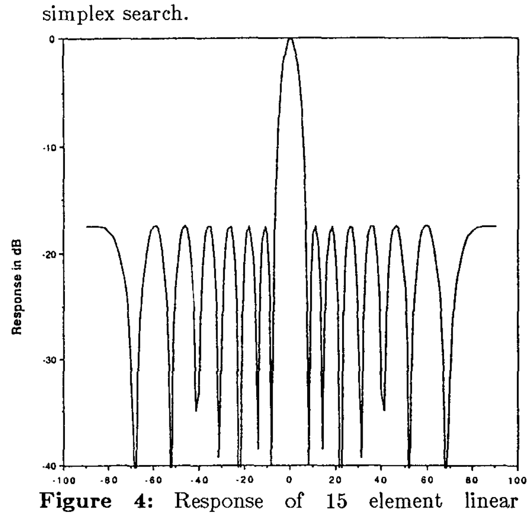

Figure 4: Response of 15 element linear equispaced array designed using Chebyshev approximation with identical constraints as for Figs. 2 and 3.

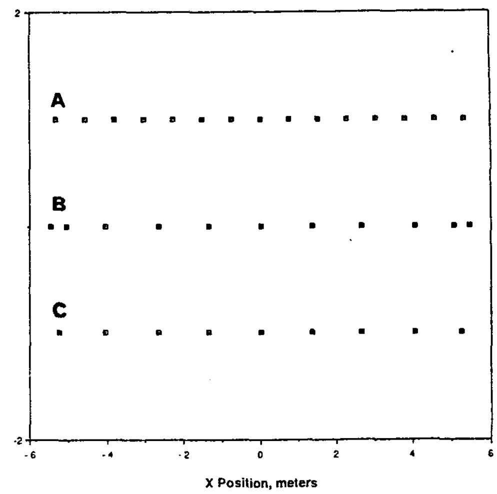

Figure 5: Array element locations for a) 15 element equispaced array (Fig. 5); b) 11 element thinned array (fig. 2); c) 9 element thinned array (fig. 3).

Circular Array Design

Consider the 60 element transparent concentric ring array of Figure 6. We wish to form beams, steered horizontally, in the plane containing the array. This is similar to the configuration used by some "dipping" sonar systems which suspend a cylindrical ring array in the water from a helicopter and form horizontal search beams. Figure 7 shows the beam response for the full array using unity magnitude shading, with complex phase shifting at each element equal to the conjugate of the elemental propagation phase delay for a plane wave arriving from the maximum response angle (MRA) of zero degrees. A sinusoidal signal at 1 kHz is assumed, which gives an average element to element spacing of just over $\lambda/4$. We require the element positions to be symmetric about the origin.

For the thinned array design, we use the same element phasing as in Figure 7, but let the algorithm adjust the real amplitude shading. The mainlobe width is constrained to be the same as Figure 7, with sidelobes no larger than the first sidelobe. Allowing some of the secondary sidelobes to come up to the level of the first allows some degree of freedom which is exploited by the algorithm to thin the array. The shaded boxes in Fig. 6 show the elements remaining after thinning using the simplex search algorithm. Fig. 8 shows the corresponding response pattern. Only 16 of the original elements are needed to maintain the original mainlobe shape and maximum sidelobe level. This result agrees with earlier observations that the outer elements of a ring array are the primary contributors to beam response.

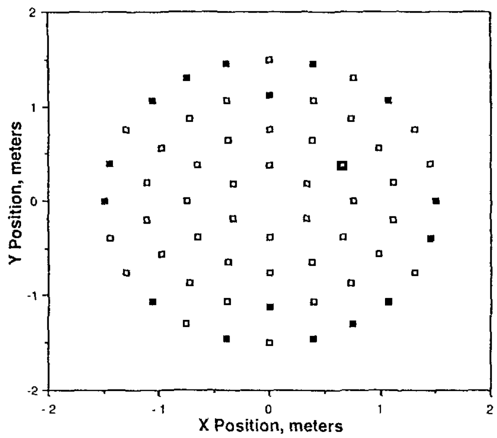

Figure 6: Spatial locations for 60 element circular array (Fig. 7). Shaded boxes indicate the 16 elements remaining after thinning (Fig. 8).

Acknowledgements

This work supported in part by the Hughes Aircraft Company Doctoral Fellowship Program.

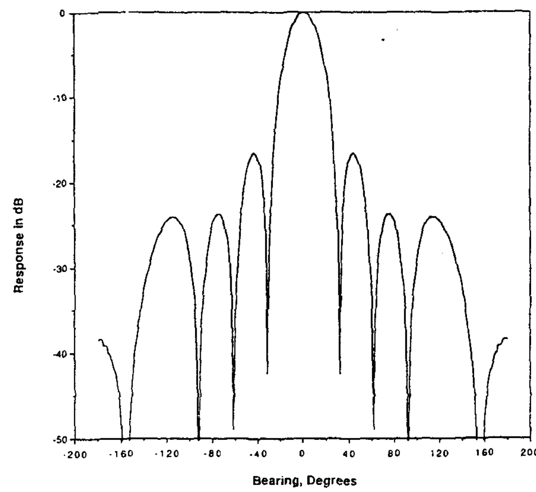

Figure 7: Response of the 60 element circular array with unit magnitude shading applied to each element.

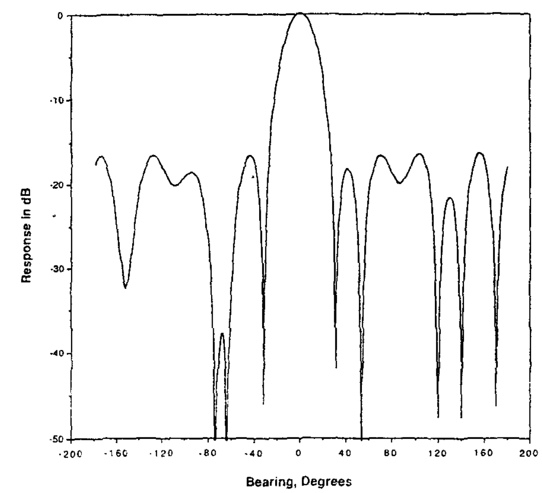

Figure 8: Response of circular array after thinning from 60 to 16 elements using local simplex search.

References

[1] A. Ishimaru and Y-S Chen, *'Thinning and broadbanding antenna arrays by unequal spacings'*, IEEE Trans. Anten. Prop., vol. AP-13, pp 34-42, 1965.

[2] P. Jarske, T. Saramaki, S. Mitra, U Neuvo, *'On Properties and Design of Nonuniformly Spaced Linear Arrays'*, IEEE Trans. Acoust., Speech, Signal Processing, vol ASSP-36, pp 372-380, 1988.

[3] T. Frank, J. Kesner and H. Gruen, *'Conformal array beam patterns and directivity indices'*, J. Acoust. Soc. Am., VOl. 63, pp 841-847, 1978.

[4] R. Streit, *'Optimization of discrete arrays of arbitrary geometry'*, J. Acoust., Soc. AM., VOl. 69, pp 199-212, 1981.

[5] E. Sullivan, *'Side-lobe behaviour of conformal arrays'*, J. Acoust. Soc. Am., Vol 71, pp 402-404, 1982.

[6] S. Prasad and R. Charan, *'On the constrained synthesis of array patterns with applications to circular and arc arrays'*, IEEE Trans. Anten. Prop., Vol AP-32, pp 725-730, 1984.

[7] L. Griffiths and C. Jim, *'An alternative approach to linearly constrained beamforming', IEEE Trans. Anten. Prop., Vol. AP-30, pp 27-34, 1982.*

[8] R. Hooke and T. Jeeves, *'Direct search solution of numerical and statistical problems'*, J. ACM., Vol. 8, pp 212-229, 1961.

[9] D. Luenberger, *'Linear and nonlinear programming'*, 2nd Ed., Addison Wesley, 1984.

[10] R. Leahy and B. Jeffs, *'Maximally sparse optimization for array beamforming and other applications'* submitted to IEEE Trans. Acoust. Speech Signal Proc., Sept., 1988.

[11] R. Leahy, B. Jeffs and Z. Wu, *'A DSP algorithm for minimum order solutions'*, Proc. 21st Asilomar Conf. Signals, Syst. Comp., Nov. 1987.

[12] H. Kruse, *'Degeneracy graphs and the neighbourhood problem'*, Lecture notes in Econ. Math. Systems, Vol. 260, Springer-Verlag, 1986.

[13] D. Mitra, F. Romeo and A. Sangiovanni-Vincentelli, *'Convergence and finite-time behaviour of simulated annealing'*, Adv. Appl. Prob., Vol. 18, pp 747-771, 1986.